

提示：选择题可能为多选题也可能为单选题，自行判断。

以下题目写完后自行使用电脑验证并且订正，倘若觉得自己还没有掌握，再问我要习题

合法语句练习

1. 以下哪些是合法语句? ()
A.a; B.123; C.; D.'a';
2. 以下哪些是合法语句? ()
A."123123"; B.1+2; C.+; D.(1 * 3);
3. 以下哪些是合法语句? ()
A.(); B.\n; C.int a; a; D.int a,b = 1;
4. 以下哪些是合法语句? ()
A.int a,int b; B.int a,b = 1,c = 2; C.1 + 2 == 3; D.1 > 2;
5. 以下哪些是合法语句? ()
A.if(); B.if(1){}; C.if(1 > 2); D.if(1 == 2){ 123; };
6. 以下哪些是合法语句? ()
A.int a = 1; a = -a; B.int a = 1,b = 2; a = a + -b; C.int a = 1,b = 2; a = a - +b; D.printf("123");
7. 以下哪些是合法语句? ()
A.printf("123"); B.printf('123'); C.printf("%d123",123); D.printf("%d %d",123,12 + 1);
8. 以下哪些是合法语句? ()
A.int i = 1,2,3; B.int i = 1 , printf("123"); C.printf("%d@@",123); D.printf("%d@%d",123,12 + 1);
9. 以下哪些是合法语句? ()
A.printf("1 + 2 = %d",1 + 2); B.printf("%d",1 + -2); C.printf("%d %d %d",1,2); D.printf("%d",1,2,3);
10. 以下哪些是合法语句? ()
A.for(; 1 ;); B.for(;;){} C.for(123 ; 123 ; 123); D.for(printf("123" ; printf("456"); printf("789"));
11. 以下哪些是合法语句? ()
A.int a = 'A'; B.char a = 123; C.char a = '123'; D.bool s = 1;

输入输出练习

选择题

1. 输入123@456,789 定义整形变量a,b,c如何把123放到a中，456放到b中，789放到c中？以下哪些选项正确()
A scanf("%d@%d,%d",a,b,c); B scanf(@,"a,b,c) C scanf("%d%c%d%c%d",a,b,c)
D scanf("%d%c%d%c%d",a,e,b,f,c);
2. 输入123.456.789 定义整形变量a,b,c如何把123放到a中，456放到b中，789放到c中？以下哪些选项正确()
A scanf("%d@%d,%d",a,b,c); B scanf(@,"a,b,c) C scanf("%d%c%d%c%d",a,b,c)
D scanf("%d%c%d%c%d",a,e,b,f,c);
3. 输入123.456.789 定义整形变量a。单浮点数变量b 如何把123.456放到a中，789放到b中？以下哪些选项正确()
A scanf("%f.%d",a,b); B scanf("%f%d",a,b) C scanf("%f.%f",a,b) D scanf("%d.%f",a,b);

4. 想要在运行后输出123@456以下哪些选项正确()

- A.int a = 123; printf("%d@456",a);
- B.int a = 123,b = 456;printf("%d@%d",a,b);
- C.printf("%d@%d",123,456);
- D.printf("%d%c%d",123,'@',456);

5. 想要在运行后输出 123.123@456 以下哪些选项正确()

- A.int a = 123.123; printf("%d@456",a);
- B.double a = 123.123,b = 456;printf("%lf@%d",a,123 + 333);
- C.double a = 123.123,b = 456.000000;printf("%lf@%.0lf",b);
- D.printf("%d.123@456",123);
- E.printf("%d%d%d.%d%d%d@%d",1,2,3,1,2,3,456);
- F.printf("%d %d %d",123,123,456);
- G.printf("%d.123@%d",'123','456');
- H.char a = 123,b = 456; printf("%d.123@%d",a,b);
- I.char a = 123,b = 456; printf("%d.123@%d",(int)a,(int)b);
- J. printf("123\".123@456");
- K.printf("123\".123@456");

6. 想要在屏幕上输出

```
123
456
```

以下哪些选项正确()

- A.printf("%d\\n%d",123,456);
- B.printf("123\\n%d",456);
- C.int a = 123,b = 456; printf("%d\\n%d",a,b);
- D.printf("%d",123); printf("\\n"); printf("%d",456);

数组的使用

填空题

1. int a[10]; 怎么读? 怎么使用才合法?
2. int a[10][10]; 怎么读? 怎么使用才合法?
3. 请问 int a[10], a[0]是否就是一个整形变量?

原反补码练习

选择题

1. (100001)原, 以下哪些选项与之相等 ()
 - A.(111110)反
 - B.(011110)反
 - C.(111111)补
 - D.(100001)反
 - E.(100001)补
 - F.(011111)补
2. (111111)原, 以下哪些选项与之相等 ()
 - A.(111110)反
 - B.(011110)反
 - C.(111111)补
 - D.(100000)反
 - E.(100001)补
 - F.(011111)补
3. (100001)补, 以下哪些选项与之相等 ()
 - A.(111110)反
 - B.(100000)反
 - C.(111111)原
 - D.(111110)原
 - E.(011110)原
 - F.(111111)反
 - G. (011110)反

4. (100001)反, 以下哪些选项与之相等 ()
A.(111110)原 B.(011110)原 C.(011110)补 D.(100000)补
E.(111111)原 F.(100001)原 G.(100000)补
5. (011110)补, 以下哪些选项与之相等 ()
A.(011110)原 B.(011110)反 C.(100001)原 D.(100001)反
E.(011111)原 F.(011101)反 G.(011111)反
6. (011110)反, 以下哪些选项与之相等 ()
A.(011110)原 B.(011110)补 C.(100001)补 D.(100001)反
E.(011111)补 F.(011101)反 G.(011111)反
7. (011110)原, 以下哪些选项与之相等 ()
A.(011110)补 B.(011110)反 C.(100001)补 D.(100001)反
E.(011111)补 F.(011101)反 G.(011111)反

填空题

1. (1000001)原转换成十进制等于 ()
2. (1000001)反转换成十进制等于 ()
3. (1000001)补转换成十进制等于 ()
4. (0010011)原转换成十进制等于 ()
5. (0010011)反转换成十进制等于 ()
6. (0101010)补转换成十进制等于 ()
7. (1101101)原转换成十进制等于 ()
8. (1101101)反转换成十进制等于 ()
9. (1101101)补转换成十进制等于 ()
10. (1011001)原转换成十进制等于 ()
11. (1101011)补转换成十进制等于 ()
12. (1000111)反转换成十进制等于 ()

转换题

1. 十进制下的123可以转换成原码, 反码, 补码成什么样? (合理即可)
2. 十进制下的-123可以转换成原码, 反码, 补码成什么样? (合理即可)
3. 十进制下的0可以转换成原码, 反码, 补码成什么样? (合理即可)
4. 十进制下的-53可以转换成原码, 反码, 补码成什么样? (合理即可)

计算题

前言计算机中的数字很大, 所以我们一般用科学计数法表示大概的数字, 例如 10^9 我们用科学计数法就是 $1e9$, 5×10^4 我们用科学计数法就是 $5e4$, 15313214,用科学计数法我们可以约等于 $1.5e7$, -1223456可以写成 $-1.2e6$

1. 整形类型为 () 个字节, () 位, 能表示的准确的数据范围为()//可以使用计算机, 用科学计数法可以表示为()
2. 长整形类型为 () 个字节, () 位, 能表示的准确的数据范围为()//可以使用计算机, 用科学计数法可以表示为()
3. 布尔类型为 () 个字节, () 位, 能表示的准确的数据范围为()//可以使用计算机, 用科学计数法可以表示为()
4. 字符类型为 () 个字节, () 位, 能表示的准确的数据范围为()//可以使用计算机, 用科学计数法可以表示为()
5. `char a = 1234;` 请问溢出了吗? 倘若溢出了最后的十进制答案为多少? 禁止使用代码, 自己计算
6. `char a = -1234;` 请问溢出了吗? 倘若溢出了最后的十进制答案为多少? 禁止使用代码, 自己计算
7. `bool a = -128;` 请问溢出了吗? 倘若溢出了最后的十进制答案为多少? 禁止使用代码, 自己计算
8. `int a = 1e10;` 请问溢出了吗? 倘若溢出了最后的十进制答案为多少? 禁止使用代码, 自己计算
9. `long long a = 12345 * 100000;` 请问溢出了吗? 倘若溢出了最后的十进制答案为多少? 禁止使用代码, 自己计算
10. `long long a = (long long) 12345 * 100000;` 请问溢出了吗? 倘若溢出了最后的十进制答案为多少? 禁止使用代码, 自己计算
11. `char a = 12 + 'A';` 请问溢出了吗? 倘若溢出了最后的十进制答案为多少? 禁止使用代码, 自己计算
12. `int a = 10, b = 3; printf("%d", a / b);` 对吗? 如果正确输出什么?
13. `int a = 10, b = 3; printf("%lf", a / b);` 对吗? 如果正确输出什么?
14. `int a = 10, b = 3; printf("%lf", 1.0 * a / b);` 对吗? 如果正确输出什么?
15. `int a = 10, b = 3; printf("%lf", 1.0 * a / (5 / 2));` 对吗? 如果正确输出什么?

技巧题，编程题

1. 给出两个整数 a, b 如何判断 a能够整除b?
2. 给出一个整数 a, 写出能够判断他是奇数偶数的多种方法。提示：左移右移。

分支语句相关知识

查看书本上的优先级计算以下

填空题

1. printf("%d", 1 > 2); 输出什么? ()
2. printf("%d", 1 + 1 > 2); 输出什么? ()
3. printf("%d", 1 >= 2); 输出什么? ()
4. printf("%d", 1 <= 2); 输出什么? ()
5. printf("%d", 1 == 2); 输出什么? ()
6. printf("%d", 1 >= 2 > 3 > 0); 输出什么? ()
7. printf("%d", 1 <= 2 > 2 <= 0); 输出什么? ()
8. printf("%d", 1 && 2 > 1 && 3); 输出什么? ()
9. !0 输出什么?
10. !-100 输出什么?
11. !111 输出什么?
12. 2 >> 1 输出什么?

逻辑题

1. 以下两个公式答案一定相同吗?

a,b,c可能为任意数字
a && b && c
(a && b) && (b && c)

2. 以下两个公式答案一定相同吗?

a,b,c可能为任意数字
a || b || c
(a || b) || (b || c)

3. 以下两个公式答案一定相同吗?

a,b,c可能为任意数字
a && b || c
a || c && b

4. 以下两个公式答案一定相同吗?

a,b,c可能为任意数字

a && (b || c)

a || c && b

循环语句

标准题

请写出来for循环3个语句，以及大括号里的分别做了什么事情，做多少次，什么时候做。（利用上课说过的运动会）

逻辑题

```
int sum = 0;
for(int i = 0; i <= 4; i = i + 1){
    sum = sum + 1;
    printf("%d ",sum);
}
```

第一次做的事情是先判断i是否小于等于4 然后给整形变量sum加上1然后输出sum，最后给i加上1，此时i = 1, sum = 1

第二次做的事情是先判断i是否小于等于4 然后给整形变量sum加上1然后输出sum，最后给i加上1，此时i = 2, sum = 2

第三次做的事情是先判断i是否小于等于4 然后给整形变量sum加上1然后输出sum，最后给i加上1，此时i = 3, sum = 3

第四次做的事情是先判断i是否小于等于4 然后给整形变量sum加上1然后输出sum，最后给i加上1，此时i = 4, sum = 4

第五次做的事情是先判断i是否小于等于4 发现大于4退出

最后输出 1 2 3 4 5

阅读以上代码，通过下面的中文解析以下代码。将每一次循环做的事情，以及变量的是多少写出来(若有多次且有规律则只写4次即可)。最后判断代码在做什么事情。

1.

```
int sum = 0;
for(int i = 1; i <= n; i++){
    if(n % i == 0){
        sum = sum + 1;
    }
}
if(sum >= 3){
    printf("YES");
}else printf("NO");
```

2.

```
int sum = 0;
for(; i != 1; i = i / 2){
    sum++;
}
printf("%d",sum);
```

3.

```
int a[100];
for(int i = 1; i <= 99; i++){
    scanf("%d",&a[i]);
}
for(int i = 99; i >= 1; i--){
    printf("%d ",a[i]);
}
```

4.

```
int x = 456321;
for(;x > 0; x /= 10){
    printf("%d ",x % 10);
}
```

5.

```
int x = 456321;
int a[10];
int i = 0;
for(;x > 0; x /= 10){
    a[++i] = x % 10;
}
for(int j = i; j >= 1; j--){
    printf("%d",a[j]);
}
```

6.

```
for(int i = 1; i <= 100; i++){
    int sum = 0;
    for(int j = 1; j <= i; j++){
        if(i % j == 0) sum++;
    }
    if(sum == 2){
        printf("%d ",i);
    }
}
```

```
7.  int n = 123;
    int cnt = 0;
    int a[100];
    for(; n != 0; n >= 1){
        a[++cnt] = n & 1;
    }
    for(int i = cnt; i >= 1; i--){
        printf("%d",a[i]);
    }
```

操作符的使用

填空题

```
1.  int a = 1;
    int b = ++a;
    int c = a++;
    int d = a--;
    int e = --a;
    printf("%d %d %d %d %d",b,c,d,e,a);
```

以上代码分别输出什么？

2. $\sim(10001001)$ 补 = () 十进制
3. $\sim(1101011)$ 补 = () 十进制
4. $\sim(11101)$ 补 = () 十进制
5. $\sim(11111111)$ 补 = () 十进制
6. $\sim(00001011)$ 补 = () 十进制
7. $! 1 = (\quad)$, $! -1 = (\quad)$, $! 0 = (\quad)$, $! 100000 = (\quad)$, $! -1234156 = (\quad)$

```
8.  int b = (2 > 1)? 1 : 2;
    int c = (2 != 1)? 1 + 2: 2 - 1;
    int d = (1 + 3 > 3)? 4 : 5;
    printf("%d %d %d",b,c,d);
```

以上代码输出什么？

9. (1010101) 补 & (1111001) 补 = ()补 = () 十进制
10. (001110) 补 & (11001) 补 = ()补 = () 十进制
11. (010101) 补 & (11101) 补 = ()补 = () 十进制
12. (1110101) 补 & (00000) 补 = ()补 = () 十进制
13. (1111111) 补 & (1000) 补 = ()补 = () 十进制
14. (1010101) 补 | (1111001) 补 = ()补 = () 十进制
15. (001101) 补 | (1111001) 补 = ()补 = () 十进制
16. (1111) 补 | (00001) 补 = ()补 = () 十进制
17. (01010101) 补 | (1010001) 补 = ()补 = () 十进制
18. (010101) 补 | (01010101) 补 = ()补 = () 十进制
19. (1010101) 补 ^ (1111001) 补 = ()补 = () 十进制

20. $(001101)_2 \oplus (1111001)_2 = (\quad)_2 = (\quad)_{10}$
21. $(1111)_2 \oplus (00001)_2 = (\quad)_2 = (\quad)_{10}$
22. $(01010101)_2 \oplus (1010001)_2 = (\quad)_2 = (\quad)_{10}$
23. $(010101)_2 \oplus (01010101)_2 = (\quad)_2 = (\quad)_{10}$
24. $(100)_{10} \& (123)_{10} = (\quad)_{10} = (\quad)_{10}$
25. $(111)_{10} \& (-12)_{10} = (\quad)_{10} = (\quad)_{10}$
26. $(100)_{10} \& (0)_{10} = (\quad)_{10} = (\quad)_{10}$
27. $(100)_{10} \& (100)_{10} = (\quad)_{10} = (\quad)_{10}$
28. $(100)_{10} \& (-123)_{10} = (\quad)_{10} = (\quad)_{10}$
29. $(-55)_{10} \& (-23)_{10} = (\quad)_{10} = (\quad)_{10}$
30. $(100)_{10} \& (1)_{10} = (\quad)_{10} = (\quad)_{10}$
31. $(100)_{10} \& (2)_{10} = (\quad)_{10} = (\quad)_{10}$
32. $(100)_{10} | (123)_{10} = (\quad)_{10} = (\quad)_{10}$
33. $(100)_{10} | (-133)_{10} = (\quad)_{10} = (\quad)_{10}$
34. $(-100)_{10} | (-15)_{10} = (\quad)_{10} = (\quad)_{10}$
35. $(-37)_{10} | (14)_{10} = (\quad)_{10} = (\quad)_{10}$
36. $(0)_{10} | (10000)_{10} = (\quad)_{10} = (\quad)_{10}$
37. $(10000)_{10} \& (10000)_{10} = (\quad)_{10} = (\quad)_{10}$
38. $(100)_{10} \wedge (123)_{10} = (\quad)_{10} = (\quad)_{10}$
39. $(100)_{10} \wedge (1)_{10} = (\quad)_{10} = (\quad)_{10}$
40. $(100)_{10} \wedge (100)_{10} = (\quad)_{10} = (\quad)_{10}$
41. $(10000)_{10} \wedge (0)_{10} = (\quad)_{10} = (\quad)_{10}$
42. $(-32)_{10} \wedge (-15)_{10} = (\quad)_{10} = (\quad)_{10}$
43. $(-37)_{10} \wedge (153)_{10} = (\quad)_{10} = (\quad)_{10}$
44. $(-1000000)_{10} \wedge (-37)_{10} \wedge (153)_{10} \wedge (-1000000)_{10} = (\quad)_{10} = (\quad)_{10}$
45. $(-100000)_{10} \& (1000000)_{10} \& (123124124)_{10} \& (0)_{10} = (\quad)_{10} = (\quad)_{10}$
46. $(-100000)_{10} \& (1000001)_{10} \& (123124121)_{10} \& (1)_{10} = (\quad)_{10} = (\quad)_{10}$
47. $(1)_{10} | (2)_{10} | (4)_{10} | (256)_{10} | (1024)_{10} = (\quad)_{10} = (\quad)_{10}$
48. $(255)_{10} | (123)_{10} | (111)_{10} | (42)_{10} | (12)_{10} = (\quad)_{10} = (\quad)_{10}$
49. $(255)_{10} \& (127)_{10} \& (63)_{10} \& (31)_{10} = (\quad)_{10} = (\quad)_{10}$
50. $(123)_{10} \& (18)_{10} \wedge (10001221)_{10} | (372717)_{10} \& (0)_{10} = (\quad)_{10} = (\quad)_{10}$
51. $(12783)_{10} \& (12323)_{10} \wedge (234)_{10} \wedge (256)_{10} | (-1)_{10} = (\quad)_{10} = (\quad)_{10}$

52. $(123)_{10} \& (127)_{10} \wedge (-123456)_{10} \wedge (-123456)_{10} = (\quad)_{10}$ 补 = $(\quad)_{10}$
53. $(-777)_{10} \wedge (-777)_{10} \mid (100)_{10} \& (100)_{10} = (\quad)_{10}$ 补 = $(\quad)_{10}$
54. $123 \gg 1 = (\quad)_{10}$
55. $123 \gg 7 = (\quad)_{10}$
56. $1024 \gg 10 = (\quad)_{10}$
57. $123 \ll 2 = (\quad)_{10}$
58. $124 \gg 3 = (\quad)_{10}$
59. $15 \ll 4 = (\quad)_{10}$
60. $2 \ll 10 = (\quad)_{10}$
61. $2 \ll 20 = (\quad)_{10}$
62. $2 \ll 25 = (\quad)_{10}$
63. $2 \ll 17 = (\quad)_{10}$
64. $1 \gg 1 = (\quad)_{10}$
65. $123 \gg 0 = (\quad)_{10}$
66. $123 \ll 0 = (\quad)_{10}$
67. $0 \ll 123 = (\quad)_{10}$
68. $(123 \gg 1) \& 1 = (\quad)_{10}$
69. $(172 \gg 3) \& 1 = (\quad)_{10}$
70. $(123213 \gg 30) \& 1 = (\quad)_{10}$
71. $(123 \ll 10) \& 1 = (\quad)_{10}$
72. $(283 \ll 10) \& 1 = (\quad)_{10}$
73. $(382 \gg 2) \mid 3 = (\quad)_{10}$
74. $(123 \ll 10) \wedge 1 = (\quad)_{10}$
75. $123 \wedge 1 = (\quad)_{10}$

技巧题

1. 写出判断整数x是否为奇数偶数的方法
2. 对于十进制数 x 的每一位我们如何取到?
3. 对于十进制数 x 转成补码的每一位我们如何取到?
4. 如何对于十进制数 x 用左移右移是的 x 变成 $10 * x$

地址

选择题

1. 一个大小为10的字符数组，以下哪个操作是能够成功输入的（ ）
A scanf("%c",&a[1]); B scanf("%c",a + 1) C scanf("%c",&a + 1) D scanf("%c",&a[5] + 1);
2. 一个大小为10的字符数组，以下哪个操作是能够成功输入的（ ）
A scanf("%s",&a); B scanf("%s",a + 1) C scanf("%s",&a + 1) D scanf("%s",&a[5] + 1);
3. 一个大小为10的整形数组，以下哪个操作是能够成功输入的（ ） |
A scanf("%d", &a + 0); B scanf("%d",&a + 3); C scanf("%d",&a[1] + 2);
D scanf("%d",a[5]);

思考题

1. int a = 2; printf("%d %d %d",a,&a,&a + 2); (假设a的首地址为0，并且不考虑越界问题，请写出预期输出值)
2. int a[10] = {0,1,2}; printf("%d %d %d",a,a + 3,a + 5); (假设a的首地址为0，请写出预期输出值)
3. double a[5] = {1.2}; printf("%d %d %.1lf",a + 3,&a,a[0]) (假设a的首地址为0，请写出预期输出值)
4. 若a的首地址为0，下面代码预期值应该是多少？

```
int a[2] [2] = {};  
printf("%d %d %d %d %d %d %d %d %d %d",  
a,a + 1,&a[0],&a[0] + 1,a[0],a[0] + 1,a[0] [0],a[0] [0] + 1,&a[0] [0],&a[0]  
[0] + 1);
```

```
int a[4] [4] = {};  
printf("%d %d %d %d %d %d %d %d %d %d",  
a,a + 1,&a[2],&a[2] + 1,a[2],a[2] + 1,a[1] [0],a[2] [1] + 1,&a[1] [2],&a[3]  
[1] + 1);
```

```
char a[2] [3] = {'a'};  
printf("%d %d %d %d %d %d %d %d %d %d",  
a,a + 1,&a[2],&a[2] + 1,a[2],a[2] + 1,a[1] [0],a[2] [1] + 1,&a[1] [2],&a[3]  
[1] + 1);
```

5. 分别写出以下输出值

```
char a[10] = {'a','b','c'};  
scanf("%s",a + 3); //输入12345  
printf("%s",a);
```

```
char a[2][10] = {'a','b','c'};  
scanf("%s",&a[1]); //输入12345  
printf("%s %c",a[1],a[1][2]);
```

```
int a[10][3][4] = {};  
scanf("%d",&a[1][2][3]); //输入14  
printf("%d ",a[1][2][3]);  
scanf("%d",a[1][2] + 2); //输入15  
printf("%d ",a[1][2][3]);  
scanf("%d",&a[1] + 2)[2][3]); //输入16  
printf("%d",a[3][2][3]);
```